Multiple LDPC decoding for distributed source coding and video coding

Forchhammer, Søren; Luong, Huynh Van; Huang, Xin

Published in:
Proceedings of WITMSE 2011

Publication date:
2011

MULTIPLE LDPC DECODING FOR DISTRIBUTED SOURCE AND VIDEO CODING

Søren Forchhammer, Huynh Van Luong and Xin Huang

DTU Fotonik, Technical University of Denmark, Building 343, Lyngby 2800, Denmark
Email: {sofo, hulu, xhua}@fotonik.dtu.dk

ABSTRACT

Distributed source coding (DSC) is a coding paradigm for systems which fully or partly exploit the source statistics at the decoder to reduce the computational burden at the encoder. Distributed video coding (DVC) is one example. This paper considers the use of Low Density Parity Check Accumulate (LDPCA) codes in a DSC scheme with feed-back. To improve the LDPC coding performance in the context of DSC and DVC, while retaining short encoder blocks, this paper proposes multiple parallel LDPC decoding. The proposed scheme passes soft information between decoders to enhance performance. Experimental results on DVC show that the LDPCA perfomance implies a loss compared to the conditional entropy, but also that the proposed scheme reduces the DVC bit rate up to 3.9% and improves the rate-distortion (RD) performance of a Transform Domain Wyner-Ziv (TDWZ) video codec.

1. INTRODUCTION

Distributed source coding as e.g. distributed video coding [1] proposes to fully or partly exploit the redundancy at the decoder, rather than at the encoder. The Slepian-Wolf theorem [2] states, it is possible to achieve the same rate by independently encoding but jointly decoding two statistically dependent signals as for typical joint encoding and decoding (with a vanishing error probability). The Wyner-Ziv theorem [3] extends the Slepian-Wolf theorem to the lossy case, becoming the theoretical basis for DSC, where source data X are lossy coded and decoded based on a correlated source Y at the decoder.

The source data X may be predicted using the side information Y at the decoder and thereafter the predicition errors may be corrected using an error-correcting code. The coding efficiency of the error correcting code, an LDPC Accumulate (LDPCA) codec [8] in this paper, plays a key role in distributed source coding. The scheme we consider utilizes feed-back from the decoder to the encoder. To improve the performance, a Wyner-Ziv codec with multiple LDPCA decoders is proposed in this work. The proposed scheme is inspired by the work in [9] using joint bitplane LDPC decoding. Different from [9], the proposed Wyner-Ziv codec utilizes multiple LDPCA decoders in parallel and passes soft information between decoders. The modifications only involve the buffer part and the decoder, while the LDPCA encoder is not changed. The objective is to increase performance by modifying the decoder, while using the same (short) encoding blocks for low-complexity and to allow for fairly fine granularity for adaptive updating of the decoder estimates.

2. DISTRIBUTED SOURCE CODING

Based on work on distributed video coding, we shall outline one approach to distributed source coding, which codes the data X given the side information Y. The distributed video coder (TDWZ codec) will be described in Section 4. Here we note that the problem is lossy coding of coefficients of the source based on side information (key frames in DVC). The coefficients are quantized and thereafter they are decomposed into bitplanes, which are fed to a rate-compatible LDPCA encoder [8] starting from the most significant bitplane (MSB) to least significant bitplane (LSB). For each encoded bitplane, the corresponding accumulated syndrome is stored in a buffer together with an 8-bit Cyclic Redundancy Check (CRC). The decoder requests bits through a feedback channel as shown in Figure 1. We shall use the terms frames, bands, coefficients and bit-planes from DVC, where coefficents just refer to the (possibly transformed) values we want to code and bitplanes refer to any collection of source bits of the same significance as, e.g. MSB and LSB from a given set of coefficients. A band is a set of coefficients, e.g. a frequency band and finally a frame is a set of bands forming an instance of X, e.g. an image frame.

At the decoder, the side information Y is used to predict the value of X and a corresponding noise residue, which expresses the conditional probabilities (*Pr*) fed to the LDPC decoder for each bitplane. In our DSC scheme, we predict the coefficient values and the residual error may be modelled by a LaPlacian distribution. Thereafter the LDPCA decoder starts to decode the various bitplanes, ordered from MSB to LSB, to correct the bit errors [4]. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded.

For the LDPCA decoding, a Belief-Propagation (BP) algorithm is used to retrieve each transmitted bitplane. The BP algorithm is a soft-decoding approach, which passes a Log-Likelihood Ratio (LLR) of *Pr* back and forth between source nodes and the syndrome nodes. Let $X=(b_{m-1},\ldots, b_1, b_0)$ denote a quantized coefficient of a Wyner-Ziv frame, where b_{m-1} is an MSB bit and b_0 is an LSB bit and Y denotes a quantized coefficient of the side information. The LLR of a bit b_i ($0 \le i \le m$-1) of the i^{th} significant bitplane is described as:

$$L(b_i) = \log\left(\frac{\Pr(b_i = 0 \mid Y, b_{m-1}, \ldots, b_{i+1})}{\Pr(b_i = 1 \mid Y, b_{m-1}, \ldots, b_{i+1})}\right) \qquad (1)$$

where $b_{m-1} \ldots b_{i+1}$ represent bits from previous successfully decoded bits of the transformed coefficient. The decoder utilizes information from previous successfully decoded bitplanes to calculate soft information for the future bitplanes.

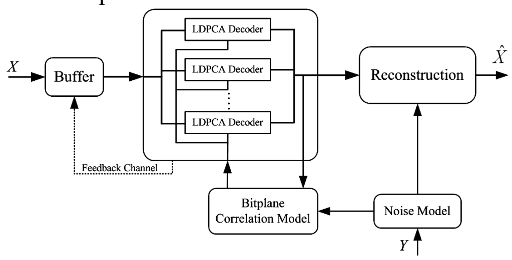

Figure 1. Multiple LDPCA Decoders

3. WYNER-ZIV CODEC WITH MULTIPLE LDPCA DECODERS

In the DSC codec described in Section 2, the LDPCA decoder utilizes side information, modeled noise correlation and the information from previous decoded bitplanes to decode future bitplanes. From the experiments it is clear that the LDPCA coding requires more bits than expressed by the conditional entropy $H(X|Y)$. A limited (short) length of the (en-)coding blocks may be desirable to retain complexity and allow for adapting the noise model leading to the conditional entropies.

As one approach to improve the performance of the LDPCA codec, a decoder may iteratively exchange information between the decoding processes of the bitplanes and refine the soft-input for each bitplane during the decoding process. Thus, a Wyner-Ziv codec with multiple LDPCA decoders is proposed. The multiple LDPCA decoders are running in parallel to keep refining the soft-input in each iteration. Each LDPCA decoder operates on the syndromes for one bitplane, but the correlation between bitplanes is exploited by passing beliefs from one bitplane to another. Once a bitplane is successfully decoded, the corresponding LDPCA decoder no longer requests syndrome bits from the buffer and the rest of the LDPCA decoders are reinitialized.

The proposed Wyner-Ziv codec using multiple LDPCA decoders is depicted in Figures 1 and 2. Soft information is exchanged between the LDPCA decoders using the so-called bitplane correlation model to reform soft-input based on feedback from the LDPCA decoders and the estimated noise distribution from the noise model. The new soft-input information of the source X is estimated and updated, expressing X-Y using the Laplacian parameter calculated by the noise model.

The main difference between this approach and [4] is that the LLR of a bit b_i ($0 \le i \le m-1$) of the i^{th} significant bitplane is computed conditional on the binary distributions (β_k, $1- \beta_k$) of bits of the other bitplanes, b_k ($k \ne i$). This means that the LLR is calculated by using soft information from the other bit-planes. Let $\beta_k = Pr(b_k=0)$ denote a probability of bitplane k. The LLR described in (1) is here generalized for a bit b_i of bitplane i as:

$$L(b_i) = \log\left(\frac{\Pr(b_i = 0 \mid Y, \beta_{m-1}, \ldots, \beta_{i+1}, \beta_{i-1}, \ldots, \beta_1, \beta_0)}{\Pr(b_i = 1 \mid Y, \beta_{m-1}, \ldots, \beta_{i+1}, \beta_{i-1}, \ldots, \beta_1, \beta_0)}\right) \qquad (2)$$

where β_k are soft values for the same coefficient as b_i.

The method involves both bitplane (bit) and coefficient (symbol) levels to update soft side information via one BP algorithm. Similar to [9], the key idea is to use the BP mechanism during the decoding of a frame and to convert the LLR back and forth between symbol level and bit level. Distinctly, in the proposed method, the soft-input is only updated after the multiple LDPCA decoders of one coefficient band are completely processed (using a certain number of iterations) at bit level based on the given syndrome bits. Let $Pr^{(t-1)}(b_k)$ denote the probability of bit b_k at the iteration t-1 at bit level. The LLR of bit b_i, is updated for iteration t as an approximation of (2):

$$L^{(t)}(b_i) = \log\left(\frac{\sum_{X \in S_0}\left(\Pr(X \mid Y)\prod_{k \ne i}\Pr^{(t-1)}(b_k)\right)}{\sum_{X \in S_1}\left(\Pr(X \mid Y)\prod_{k \ne i}\Pr^{(t-1)}(b_k)\right)}\right) \qquad (3)$$

where $X=(b_{m-1},\ldots b_i,\ldots, b_1, b_0)$, S indicates the set of values $\{0,1,2,\ldots,2^m\text{-}1\}$ for coefficient X (or its magnitude), which is coded by m bitplanes and $S_0=\{X \in S{:}b_i=0\}$, $S_1=\{X \in S{:}b_i=1\}$. $Pr(X|Y)$ is calculated at coefficient level by using the updated noise distribution between the side information coefficient and the original Wyner-Ziv coefficient via the noise model as shown in Figure 1.

The LLRs at iteration t noted by $L^{(t)}(b_i)$, are in turn input to multiple LDPCA decoders. After one LDPCA is processed, $L^{*(t)}(b_i)$ is temporarily achieved as output. The updated $Pr^{(t)}(b_i)$ values are obtained based on the LLR definition:

$$L^{*(t)}(b_i) = \log\left(\frac{\Pr^{(t)}(b_i = 0)}{1 - \Pr^{(t)}(b_i = 0)}\right) \qquad (4)$$

i.e. for the next iteration, we have:

$$\Pr^{(t)}(b_i = 0) = \frac{1}{2}\left(1 + \tanh\left(\frac{L^{*(t)}(b_i)}{2}\right)\right) \qquad (5)$$

This $Pr^{(t)}(b_i)$ is used as a new probability of bit b_i to compute new LLRs, $L^{(t+1)}(b_i)$, for the next iteration of multiple LDPCA decoding based on (3). Since all LDPCA decoders are running in parallel, once a bitplane is successfully decoded, the re-initialization procedure is performed. The new soft-inputs for the rest of the bitplanes are assigned conditional on the successfully decoded bitplane. Once a LDPCA decoder has successfully decoded a bitplane, it will no longer request syndromes from the buffer. Assume b_i is successfully decoded with value 0, then $Pr^{(t)}(b_i=0)=1$ and the iteration count is reset as $t=0$. In addition, the remaining unfinished bitplanes are re-initialized by $Pr^{(0)}(b_j=0)=1/2$. The LDPCA decoders are iteratively operated up to a maximum numbers of iterations (T_{max}) with the given syndrome bits. If they are not successful after this number of iterations, the LDPCA decoders request more syndrome bits from the buffer via the feedback channel. Then a new process is started until all the bitplanes of the current set of coefficients are successfully decoded. Let N_{max} denote a maximum numbers of syndromes.

Overall, the multiple LDPCA decoding is handled as follows:

1. **Initiate parameters**. Iteration count $t=0$; Number of syndrome bits $n=0$; For all bits b_i, $Pr^{(0)}(b_i=0)=1/2$.
2. **Increase and check conditions**.
   a. **Syndrome bit condition**: Increase $n=n+1$. If $n \geq N_{max}$ then end, else go to Step 2.b.
   b. **Iteration count condition**: Increase $t=t+1$. If $t<T_{max}$ go to Step 3, else return to 2.a.
3. **Compute the LLRs.** At bit level, (3) is computed to get the LLRs, $L^{(t)}(b_i)$.
4. **Check if any LDPCA is successfully decoded?**
   a. **No: Compute probabilities of bitplanes**. $L^{(t)}(b_i)$ are forwarded to multiple LDPCA decoders where $L^{*(t)}(b_i)$ are received from LDPCA outputs. New probabilities of bitplanes, $Pr^{(t)}(b_i)$, by (5).
   b. **Yes: Re-initialize the process**. Assume LDPCA (b_i) is successfully decoded with value $b_i=0$, assign $Pr^{(t)}(b_i=0)=1$. Reset iteration count $t=0$ and the remaining unfinished LDPCA decoders by $Pr^{(0)}(b_j=0)=1/2$;
5. **Check all LDPCA decoders**. The process is ended if all bitplanes are successfully decoded, otherwise, go to Step 2.b.

The procedure above is repeated for all bands of coefficients for which Wyner-Ziv bits are transmitted. In some cases, the length of the required syndromes consumed for the LSB is (close to) 1 bit per symbol, even though there is still some correlation. This is due to a (relative) loss in the LDPCA decoder. This may be reduced by first sending the marginalized LSB independently, as the entropy of the LSB often is close to 1 bit/symbol, and then apply multiple decoding to the remaining bitplanes after decoding the LSB and updating the soft information for the remaining bit-planes.

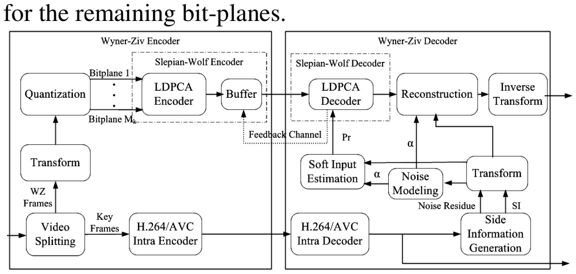

Figure 2. Architecture of feedback channel based Transform Domain Wyner-Ziv video codec

4. STATE-OF-THE-ART TRANSFORM DOMAIN WYNER-ZIV VIDEO CODING

Transform Domain Wyner-Ziv (TDWZ) video coding is a popular approach to DVC [1]. It has been improved by e.g. advanced side information generation schemes [5], finer noise models [4][5] and refinement schemes [7]. Despite the advances in practical TDWZ video coding, the RD performance still trails the performance of conventional video coding, such as H.264/AVC. The architecture of a TDWZ video codec is depicted in Figure 2. It basically follows the same architecture as the DISCOVER one [4]. However, a better side information generation scheme [5] and an improved noise model [6] are adopted. At the encoder, periodically one frame out of N in the video sequence is named as key frame and intermediate frames are WZ frames. The key frames are intra coded by using low complexity video coding as H.264/AVC Intra, while the WZ frames in between are coded with a Wyner-Ziv approach. WZ frames are transformed using a 4x4 block size and the transformed coefficients within the same frequency band are grouped together and then quantized. At the decoder, a side information frame is interpolated and the corresponding noise residue is generated by using previously decoded frames. The noise residue is modeled assuming a Laplacian distribution of $|X-Y|$. The data are decoded using single or multiple LDPCA decoding as outlined. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

5. PERFORMANCE EVALUATION

In this section, the RD performance of the proposed approach [10] is presented and compared with the state-of-the-art TDWZ video codec described in Section 4 as well as relevant benchmarks. The test sequences are 149 frames of *Foreman*, *Hall Monitor*, *Soccer*, and *Coastguard* (15Hz, QCIF). GOP (group of pictures) size is 2, where odd frames are coded as a key frame using H.246/AVC Intra and even frames are coded using Wyner-Ziv coding. Eight RD points (Q_j) are considered corresponding to eight 4x4 quantization matrices [4], which also determine the number of bitplanes, m, of each DCT coefficient band. The proposed model uses m regular LDPC accumulate decoders [8], with a length of 1584 bits each, for the 1584 transform coefficients. The m LDPCA each decodes one bitplane.

Table 1 shows rate and PSNR values of the proposed TDWZ codec with multiple LDPCA decoders (WZMD) as well as the savings in total rate, ΔR (in %), and WZ rate, ΔR_{WZ} (in %), compared with the state-of-the-art TDWZ codec [6]. The WZMD achieves a reduction of bit-rate for WZ frames up to 1.8% for *Foreman*; 2.59% for *Hall Monitor*; 2.26% for *Soccer*; 1.82% for *Coastguard*.

In some cases, the length of the required syndromes for the LSB is (close to) 1 bit per symbol, even though there is still some correlation. This LDPCA decoder loss, which may be reduced by first coding the LSB independently and thereafter apply WZMD to the remaining bitplanes. This is called WZMD(LSB). Up to three LSB bitplanes may be sent first. Deviating from distributed encoding, the Ideal Code Length ICL may be interpreted as the number of bits required by a backward adaptive prediction video coding scheme applying ideal arithmetic coding to the calculated soft-input values, Pr, which the encoder can also calculate if it duplicates the processing of the decoder. The decision is based on thresholding the

ICL for the LSB. For 1-5 bitplanes the LSB is evaluated. For 6 and 7 bitplanes, 2 and 3 LSB bitplanes are evaluated, respectively. The thresholds applied are 0.89, 0.95 and 0.98.

As a result, the coding efficiency in terms of bit-rate is improved. Table 2 depicts the WZ bit rate savings for WZMD and WZMD(LSB) compared with TDWZ [6]. The results shows that WZ rate savings up to 3.9% for *Foreman* and 3.77% for *Soccer*. In a follow-up work [11] we have included inter bitplane correlation refinement in the loop of WZMD coding for additional performance.

The experimental results in Fig. 3 demonstrate that the proposed approach significantly improves overall RD performance compared with the DISCOVER codec, with PSNR gains up to about 0.7 dB for *Foreman* and 0.9 dB for *Soccer*. The performance of H.264/AVC (Intra), the H.264/AVC (No Motion), and ICL codecs are also included. The WZMD is more efficient than H.264/AVC (Intra) for *Foreman*.

6. CONCLUSION

This paper considers LDPCA for DSC and DVC and proposes a Wyner-Ziv codec using multiple parallel LDPC decoding by passing soft information between the bitplanes during the decoding process. Experimental results show that the proposed multiple LDPC decoding can improve the coding efficiency of DVC (TDWZ) in terms of WZ rate savings up to 3.9% compared with the corresponding single LDPC TDWZ [6] coding.

7. REFERENCES

[1] B. Girod, A. Aaron, S. Rane, D. Rebollo-Monedero, "Distributed Video Coding" *Proceedings of the IEEE,* vol. 93, no. 1, pp. 71-83, January 2005.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, pp. 471–480, Jul. 1973.

[3] A.D. Wyner, J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22 (1), pp. 1–10, Jan. 1976.

[4] X. Artigas e.a, "The DISCOVER codec: architecture, techniques and evaluation," *PCS*, Lisbon, Portugal, Nov. 2007.

[5] X. Huang, S. Forchhammer, "Improved side information generation for distributed video coding," *IEEE Int'l Workshop Multimedia Signal Proc.*, Cairns, Australia, Oct. 2008.

[6] X. Huang, S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," *IEEE ICASSP*, Taipei, Taiwan, ROC. April 2009.

[7] X. Huang, S. Forchhammer, "Transform domain Wyner-Ziv video coding with refinement of noise residue and side information," *SPIE VCIP*, HuangShan, China, July 2010.

[8] D. Varodayan, A. Aaron, B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing*, vol. 23 (11), pp. 3123-3130, 2006.

[9] D. Varodayan, D. Chen, M. Flierl, B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning," *EURASIP Signal Processing, Image Communication*, vol. 23 (5), pp. 369-378, 2008.

[10] H. Luong, X. Huang, S. Forchhammer, "Multiple LDPC Decoding using Bitplane Correlation for Transform Domain Wyner-Ziv Video Coding," *IEEE ICASSP*, Prague, Czech, May 2011.

[11] H. Luong, X. Huang, S. Forchhammer, "Parallel Iterative Decoding of Transform Domain Wyner-Ziv Video using Cross Bitplane Correlation," *IEEE ICIP*, Brussels, Belgium, 2011 (accepted).

Table 1. WZ rates and rate savings (in %) for WZMD based TDWZ compared with TDWZ [6]

Q_j	Foreman				Hall				Soccer				Coast-guard			
	ICL [kbps]	Rate [kbps]	PSNR [dB]	ΔR_{WZ} [%]	ICL [kbps]	Rate [kbps]	PSNR [dB]	ΔR_{WZ} [%]	ICL [kbps]	Rate [kbps]	PSNR [dB]	ΔR_{WZ} [%]	ICL [kbps]	Rate [kbps]	PSNR [dB]	ΔR_{WZ} [%]
1	20.85	25.74	28.65	1.32	7.99	11.84	31.72	1.46	32.22	38.07	28.14	1.88	11.92	16.08	28.56	1.58
2	28.78	34.68	29.38	1.51	12.13	16.91	32.31	1.77	40.88	48.55	28.66	1.62	17.49	22.63	29.25	1.53
3	31.11	39.06	29.84	1.30	13.32	19.88	32.34	1.00	44.70	54.07	29.37	2.06	19.12	25.91	29.34	1.62
4	47.99	62.17	32.26	1.66	17.02	27.46	34.54	1.72	68.38	85.07	31.91	1.88	29.55	41.25	31.06	1.53
5	52.57	68.28	32.38	1.80	17.90	29.93	34.55	2.59	72.31	90.19	32.01	2.26	30.21	43.56	31.47	1.05
6	73.76	92.97	33.55	1.78	27.46	42.88	36.14	1.80	97.08	119.91	33.01	2.06	46.09	63.76	32.61	1.82
7	97.31	122.14	35.75	1.61	33.99	52.29	37.59	1.71	127.97	157.72	35.26	1.80	67.40	90.84	33.76	1.44
8	171.83	210.32	39.37	1.35	56.23	82.06	40.86	1.82	217.81	263.84	38.96	1.24	136.71	175.11	36.96	1.07

Table 2. Bit rate savings (in %) of WZMD and WZMD (LSB)

Q_j	Foreman				Soccer			
	WZMD		WZMD(LSB)		WZMD		WZMD(LSB)	
	ΔR [%]	ΔR_{WZ} [%]	ΔR [%]	ΔR_{WZ} [%]	ΔR [%]	ΔR_{WZ} [%]	ΔR [%]	ΔR_{WZ} [%]
1	0.49	1.32	1.44	3.90	1.26	1.88	2.51	3.77
2	0.62	1.51	1.48	3.60	1.11	1.62	1.95	2.84
3	0.53	1.30	0.99	2.41	1.38	2.06	1.89	2.82
4	0.68	1.66	0.68	1.66	1.19	1.88	1.38	2.18
5	0.78	1.80	0.78	1.80	1.46	2.26	1.62	2.51
6	0.82	1.78	1.05	2.26	1.35	2.06	1.41	2.15
7	0.73	1.61	0.86	1.89	1.14	1.80	1.29	2.03
8	0.66	1.35	0.79	1.62	0.73	1.24	0.80	1.36

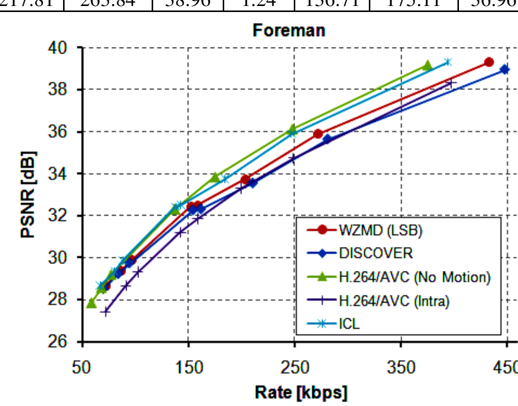

Figure 3. RD performance comparison